

## Declarative Programming

Workshop exercises set 1 (for workshops in week 2).

### QUESTION 1

What are the most annoying limitations of a programming language you have used? It would be good if you posted something on this topic to the LMS discussion forum.

### QUESTION 2

What are some useful Haskell resources on the web?

### QUESTION 3

What will be printed by this C code fragment?

```
#include <stdio.h>
int f(int x, int y)
{
    return 10 * x + y;
}

int main(void)
{
    int i, j;

    i = 0;
    j = f(++i, ++i);
    printf("%d\n", j);
    return 0;
}
```

What does this show about the impact of side effects in C?

### QUESTION 4

Fix the formatting errors (due the offside rule) in the following code, making the minimal possible changes.

```
zero = len []
one
    = len
        []
two = len [1,2] three = len [1,2,3]
four = len [1,
            2,3,
            4]
len [] = 0
len (x:xs) = 1 + len xs
```

### QUESTION 5

Implement a function to perform the logical XOR (exclusive or) operation. The XOR of two truth values is true if exactly one of them is true. You may approach this using pattern matching or using other logical operations.

### QUESTION 6

Implement a function to append two lists in Haskell (this is the ++ function in the standard prelude). What is the type

of this function?

#### QUESTION 7

Implement your own version of the 'reverse' function included in the Haskell Prelude. Do not use the existing 'reverse' function in your implementation. Note you should call your function 'myReverse' to avoid shadowing the existing function.

#### QUESTION 8

Implement a function 'getNthElem' which takes an integer 'n' and a list, and returns the nth element of the list.